NATAS REPORT

Level 0 → Level 1

Username: natas0
Password: natas0

Steps:

- Open Natas 0 URL in the browser.
- Right-click → View Page Source.
- Find password hidden in the HTML comment.

Tools Used: Browser

Logic:

- Sometimes information is hidden inside page comments.

Level 1 → Level 2

Username: natas1
Password: 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq

Steps:

- Open Natas 1 URL.
- Page says to find password on the page.
- Right-click → View Page Source.
- Password is inside a hidden field.

Tools Used: Browser

Logic:

- Hidden fields can store important info.

Level 2 → Level 3

Username: natas2
Password: Vf1n1klkpA1DT1yHH4uwnZqIfEKq5ZgA

Steps:

- Open Natas 2 URL.
- Right-click → View Page Source.
- Find a link to a /files/ directory.

- Browse the files and find the password.

Tools Used: Browser

Logic:

- Sometimes hidden directories contain files with sensitive info.

Level 3 → Level 4

Username: natas3
Password: XwLhJkd9FZtY9TS25wUzI1djVJdOxod1

Steps:

- Open Natas 3 URL.
- View Page Source.
- Find a hidden link to an image folder.
- Explore folders and find the password.

Tools Used: Browser

Logic:

- Hidden links inside HTML can lead to useful folders.

Level 4 → Level 5

Username: natas4
Password: xDiXqQcgVgsYVm6FvO2lkwuw3vA1wPnd

Steps:

- Open Natas 4 URL.
- Access denied without the right referer header.
- Set a fake Referer header using browser extension or Python.

Tools Used: Browser (extensions) / Python (requests)

Logic:

- Servers check headers like Referer to give access.

Level 5 → Level 6

Username: natas5
Password: oZ4doF4ebnObh9h9z1ODAtd1v3PC4Y6k

Steps:

- Open Natas 5 URL.
- Website checks cookies for login.
- Edit cookies to make logged_in=true.

Tools Used: Browser (Developer Tools → Cookies)

Logic:

- Websites often store login status in cookies.

Level 6 → Level 7

Username: natas6
Password: N9pTrpzqfyyhAvcq3Wm9Ge34TcCHAk9q

Steps:

- Open Natas 6 URL.
- Password is generated using a secret and input.
- Guess the right input or reverse the hash.

Tools Used: Browser, Python (hashlib library)

Logic:

- Hashing can be brute-forced if weak.

Level 7 → Level 8

Username: natas7
Password: u8fWScEdm7QUpgo0owWeNNvbtbnZtsfS

Steps:

- Open Natas 7 URL.
- URL uses GET parameters like page=home.
- Try changing page parameter to interesting files like /etc/natas_webpass/natas8.

Tools Used: Browser

Logic:

- URL parameters can be exploited to access hidden files.

Level 8 → Level 9

Username: natas8
Password: 5ad2b10026ad65cdb1dbf97cfa117fbc

Steps:

- Open Natas 8 URL.
- View page source.
- Password is XOR-encrypted.
- Write Python script to reverse XOR.

Tools Used: Browser, Python

Logic:

- XOR encryption can be cracked if weak or known data is available.

Level 9 → Level 10

Username: natas9
Password: f2e706f27b29a30c0b672f60d80cf875

Steps:

- Open Natas 9 URL.
- Website has search feature vulnerable to command injection.
- Inject commands to read password file.

Tools Used: Browser

Logic:

- Improper input validation can allow command injection.

Level 10 → Level 11

Username: natas10
Password: nOpp1igQAkUzaI1GUUjzn1bFVj7xCNzu

Steps:

- Open Natas 10 URL.
- It has a search form vulnerable to command injection.
- Inject a command like ; cat /etc/natas_webpass/natas11.

Tools Used: Browser

Logic:

- Input fields can allow command execution if not filtered properly.

Level 11 → Level 12

Username: natas11
Password: U82q5TCMMQ9xuFoI3dYX61s7OZD9JKoK

Steps:

- Open Natas 11 URL.
- Website uses cookies encrypted with XOR.
- Decrypt cookie using known XOR and find password.

Tools Used: Browser, Python

Logic:

- XOR encryption can be reversed if we know plaintext structure.

Level 12 → Level 13

Username: natas12
Password: EDXp0pS26wLKHZy1rDBPUZk0RKfLGIR3

Steps:

- Open Natas 12 URL.
- Website allows uploading files.
- Upload a PHP file disguised as an image.
- Execute uploaded PHP file to read password.

Tools Used: Browser, PHP

Logic:

- Improper file validation lets attackers upload and run code.

Level 13 → Level 14

Username: natas13
Password: jmLTY0qiPZBbaKc9341cqPQZBJv7MQbY

Steps:

- Open Natas 13 URL.
- Same as Level 12, but now file extension is checked.

- Upload a real JPG file with hidden PHP inside.

Tools Used: Browser, PHP

Logic:

- File content and extension must be tricked to bypass checks.

Level 14 → Level 15

Username: natas14
Password: Lg96M10TdfaPyVBkJdjymbllQ5L6qdl1

Steps:

- Open Natas 14 URL.
- Login form vulnerable to SQL Injection.
- Use `natas15" OR "1"="1` to bypass login.

Tools Used: Browser, SQLi techniques

Logic:

- Weak SQL queries can be broken using injections.

Level 15 → Level 16

Username: natas15
Password: AwWj0w5cvxrZiONgZ9J5stNVkmxdk39J

Steps:

- Open Natas 15 URL.
- SQL Injection blind attack needed (no visible output).
- Script a time-based attack or guess character by character.

Tools Used: Python (requests), Browser

Logic:

- Blind SQLi leaks info based on response time or success.

Level 16 → Level 17

Username: natas16
Password: WaIHEacj63wnNIBROHeqi3p9t0m5nhmh

Steps:

- Open Natas 16 URL.
- Site filters dangerous characters, but injection possible.
- Use pattern matching like $(grep a /etc/natas_webpass/natas17).

Tools Used: Browser, Burp Suite (optional)

Logic:

- Input sanitization might miss some tricks like pattern injections.

Level 17 → Level 18

Username: natas17
Password: 8Ps3H0GWbn5rd9S7GmAdgQNdkhPkq9cw

Steps:

- Open Natas 17 URL.
- Blind SQL Injection with time delays.
- Script guessing password using time taken by server.

Tools Used: Python (requests, time module)

Logic:

- If query is true, server response is delayed → can find password character-by-character.

Level 18 → Level 19

Username: natas18
Password: xvKIqDjy4OPv7wCRgDlmj0pFsCsDjhdP

Steps:

- Open Natas 18 URL.
- Site uses cookies for user ID.
- Bruteforce cookie values until finding Admin.

Tools Used: Python (loop), Browser

Logic:

- If cookie-based, enumerate IDs until admin access is granted.

Level 19 → Level 20

Username: natas19
Password: 4IwIrekcuZlA9OsjOkoUtwU6lhokCPYs

Steps:

- Open Natas 19 URL.
- Cookie values are encoded.
- Decode cookies (hex) and bruteforce for Admin.

Tools Used: Python, Hex decoder

Logic:

- Encoded data can be bruteforced when simple patterns like hex encoding are used.

Level 30 → Level 31

Username: natas30
Password: wie9iexae0Daihohv8v6uluana4i5q6e

Steps:

- Open Natas 30 URL.
- Site uses POST method.
- Send manipulated POST data to change parameters like admin=1.

Tools Used: Browser (Developer Tools → Network / POST Editor)

Logic:

- Sometimes sites trust form values without validation.

Level 31 → Level 32

Username: natas31
Password: t3bKABA8j8T8Jc9i2k4biyFtZzhT6v7Z

Steps:

- Open Natas 31 URL.
- Upload a file or send crafted POST data.
- Use the vulnerabilities to upload a PHP script and execute.

Tools Used: Browser, Burp Suite (optional)

Logic:

- Upload functions without proper checks allow code execution

Level 32 → Level 33

Username: natas32
Password: WaIHEacj63wnNIBROHeqi3p9t0m5nhmh

Steps:

- Open Natas 32 URL.
- Use SQL Injection or file inclusion by modifying parameters.
- Trick server to execute internal files and get password.

Tools Used: Browser, Python script (for faster testing)

Logic:

- SQLi + file read techniques are mixed for exploitation.

- Level 34 doesn't exist in the OverTheWire Natas series.

FINAL NOTES:

- Tools used: Browser, Python (requests), Burp Suite (optional).
- Logic: Focused on web vulnerabilities like SQL Injection, File Upload, Command Injection, and Weak Session Handling.